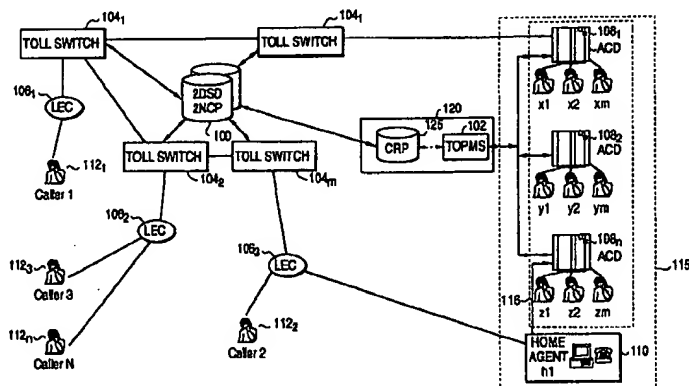




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : H04M 3/50		A1	(11) International Publication Number: WO 98/42118
			(43) International Publication Date: 24 September 1998 (24.09.98)
(21) International Application Number: PCT/US98/04872		(81) Designated States: CA, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).	
(22) International Filing Date: 16 March 1998 (16.03.98)		Published With international search report. With amended claims.	
(30) Priority Data: 60/041,077 18 March 1997 (18.03.97) US 09/040,223 12 March 1998 (12.03.98) US			
(71) Applicant: AT & T CORPORATION [US/US]; 32 Avenue of the Americas, New York, NY 10003 (US).			
(72) Inventors: DURINOVIC-JOHRI, Sanja; 20 Woodman Place, Aberdeen, NJ 07747 (US). LEVY, Yonatan, A.; 1 Grant Road, Manalapan, NJ 07726 (US). SABINAS, Suhasini, V.; 14 Cape May Drive, Marlboro, NJ 07746 (US).			
(74) Agents: JACKSON, Thomas, H. et al.; Banner & Witcoff, Ltd., 11th floor, 1001 G Street, N.W., Washington, DC 20001-4597 (US).			

(54) Title: METHOD AND APPARATUS FOR SIMULATING A CENTRAL QUEUE FOR DISTRIBUTING CALLS IN A DISTRIBUTED ARRANGEMENT OF AUTOMATIC CALL DISTRIBUTORS



(57) Abstract

Calls may be distributed and load balancing maintained for a distributed automatic call distributor (ACD) system by simulating a central FIFO queue at the central routing point of the system. The simulated FIFO queue either requires that the number of calls in queue and the oldest call waiting time of calls at each automatic call distributor be periodically provided to the central routing point or event-based data be provided on each call answered in the system or not routed by the central routing point. Deviations from FIFO order of service recorded at the simulated FIFO queue may signal the need for queueing a call at another automatic call distributor. With additional information on agents available, the call removal rate may be estimated. Also, delay estimates may be updated based on comparing actual and estimated delays over time. The routing point may be updated with traffic data through, for example, a known telemarketing operations performance management system (TOPMS) or directly from the automatic call distributors via the toll telecommunications network.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

**METHOD AND APPARATUS FOR SIMULATING
A CENTRAL QUEUE FOR DISTRIBUTING CALLS IN A
DISTRIBUTED ARRANGEMENT OF AUTOMATIC CALL DISTRIBUTORS**

This application claims priority to U.S. Provisional Patent Application Serial No. 60/041,077, filed March 18, 1997 of Sanja Durinovic-Johri.

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention relates to dynamically routing calls received into a central controller for a distributed automatic call distributor arrangement and, in particular, to method and apparatus for routing calls in accordance with a process of simulating a central first-in first-out (FIFO) queue, referred to herein as a simulated or virtual queue, whereby an estimate of the actual call removal rate (the rate at which calls are cleared from an actual queue), the estimated delays a caller will have to wait and the like, may be obtained.

2. Description of the Related Arts

Figure 2 of U. S. Patent No. 5,530,744, entitled Method and System for Dynamic Customized Call Routing, incorporated herein by reference as to its entire contents, shows a Telemarketing Operations Performance Management System (TOPMS). TOPMS collects event-based data from Automatic Call Distributors (ACDs) and forwards the data to a central Customer Routing Point (CRP). TOPMS may be regarded as a staging area for collecting traffic data from distributed ACDs and home agents and forwarding the collected data to a processor of the Customer Routing Point. The Customer Routing Point makes call routing decisions for 1-800, 1-900 and 1-888 dialed calls among others for the distributed ACD/home agent arrangement so that calls may be handled as efficiently as possible. In accordance with the '744 patent, call load management, or the loading of incoming calls to be handled by servers at distributed Automatic Call Distributor (ACD) locations and home agent sites, occurs as calls are received into the

5 system on a call-by-call basis according to centralized queue control managed by the CRP and in accordance with information received from the TOPMS with calls being queued at the ACDs. These terms, for example, TOPMS and CRP were used in the '744 patent and do not generally describe the dynamic call routing system of the '744 patent; however, the terms as used therein will be used similarly in the present application whenever possible. Again, TOPMS is a call traffic data collection system that collects all data and forms an integrated view of a customer's business application. Further details about TOPMS may be found in U.S. Patent No. 5,164,983, also incorporated herein by reference.

10 As used in the present application, a business application is used by way of example as the reason for the call, for example, a sales application, to make an airline reservation, catalog sales and the like. A CRP is the location of a computer processor for processing the collected data and making call routing decisions to servers at the distributed ACD locations and home sites. While shown in the '744 patent as separate systems, the CRP and TOPMS may be at the same location or distributed or one may share or be combined in functionality with the other.

15 Automatic call distributors (ACDs) are known switches which are capable of receiving incoming telephone calls and routing the calls to agents manning service positions, referred to herein as agent groups. Automatic call distributor switches are manufactured by several manufacturers including Lucent Technologies Inc. Others known in the art include the Definity, the Rockwell Galaxy or Spectrum, Siemens Rolm brand ACDs, Northern Telecom ACD and Aspect brand ACDs. These collect and output traffic related snapshot data indicating, for example, the number of server positions presently manned, the average call handling time, the number of calls in a queue or the number of available agents and the cumulative number of calls offered whereby, according to the '744 patent, parameters may be estimated that assist in call routing decisions. Thus, the '744 patent emphasizes snapshot data and considers historical data.

20 In U. S. Patent No. 5,546,452, there is described a communications system using a central controller to control at least one network and agent system. At column 5, lines 45-60, and with reference to Figure 2, there is described a process of taking into consideration real-time requested service data as well as historical data in making network

25

30

control decisions.

By way of example and referring to Figure 1, an airline reservation service may have a number of ACD locations $108_1, 108_2, \dots, 108_n$ dispersed about a country or even throughout the world, for example, in different time zones. Figure 1 is derived from Figure 2 of the '744 patent but has been modified in part for better describing the present invention.

Also, agents may operate out of their homes and are known as home agents 110. Only one home agent is shown. These ACD locations 108 and home agents 110 are shown in Figure 1 comprising a service system 115. Basically data is collected from these locations 108, 110 and expected delays in call waiting times are balanced among the several sites either based on expected delay times, geographic preferences or customer specified routing options. One approach might be a Minimum Expected Delay Algorithm. Another might be to include consideration of geographic preferences.

As calls enter the system to be handled by servers which may be anywhere in the country, the decision on what location to route the call to is centrally managed via a system 120 comprising TOPMS 102 and a customer routing point (CRP) processor 125. The '744 patent approach to dynamically distributing calls via an algorithm at the customer routing point 125 may be referred to herein as an agent pool based approach.

The agent pool based approach requires four or more parameters from the distributed ACDs 108 and home agents 110 of the service system 115 it serves. Unfortunately, different manufacturers of ACDs may or may not provide all these parameters for call routing decisions. These include and may not be limited to the data described above: 1) the total agent pool size at a given point in time for a particular application (such a pool size may change over time of day as ACD offices close and new offices open) 2) the average call handling time (how long typically does it take an agent to handle a call) 3) the number of calls in queue or the number of available agents, and 4) the cumulative number of calls offered (a cumulative counter of calls offered to a site) which together require several input measures to compose. For example, 1) total agent pool size requires all agents currently available, agents currently talking on inbound calls and agents occupied in post call processing as raw input measures.

The availability or busy status of any ACD agent $x_1, y_1, z_1 \dots z_m$ may be reported via the TOPMS. The drawing is simplified to suggest that m agents are on duty at each ACD location, when in fact the number of agents at each site typically differs. Availability and busy status of home agents 110 is reported via the network to system 120. This data reporting is shown, for example, in Figure 1 by way of link 116 to one of the ACD locations, which typically would be the location nearest the home agents 110. On the other hand, non-local (1-800, 1-900 and 1-888) calls are routed to the home agents upon command of the CRP 125 via Network Control Point (NCP) 100 for controlling the toll network. The CRP 125 is a typical processor comprising application algorithms including the agent-pool based call routing algorithm and memory for storing call traffic data updates forwarded via TOPMS 102.

A major issue associated with agent-pool-based algorithms is that the number of ACD agents $x_1, y_1, z_1 \dots z_m$ may be inaccurate due to sharing of agents across queues of ACDs and/or applications. This can be seen from Figure 2. For example, an airline agent may take international reservations A, domestic reservations B, first class/frequent flyer reservations C and further arrange for air freight D. Let us assume that D is a higher priority than A, C has the same priority as A and B has a lower priority than A. Four groups of agents are shown G1 - G4 each having a different specialty. G1 handles international reservations (A) while G2, for example, handles international (A) and domestic (B). Calls for international reservations can be queued in any serving line G1-G4 and when handled by one server, the call disappears from all the queues. Sharing of servers across queues complicates the problem that has to be solved by the agent-pool-based algorithm. Thus, the major issue in predicting expected delay with the agent pool based algorithm is in estimating the total agent pool size when calls can be queued among multiple queues and specialty queues as shown in Figure 2, thus introducing ambiguities.

Consequently, there remains a need in the art for a more simplified method and apparatus for dynamically routing calls in a distributed ACD environment. The agent-pool-based method is subject to the above-described inaccuracies and ambiguities, relies on the collection of too many data elements and relies on the availability of such data from different vendor ACDs. Moreover, a solution should consider both real-time and

historical data if available or collectible from the ACDs or home agents.

SUMMARY OF THE INVENTION

The problems and related problems of prior art methods of dynamically routing calls in a distributed queuing environment are solved by the principles of the present invention, a method for simulating a central FIFO queue (a virtual FIFO queue), whereby the system maintains a central queue of all the calls physically queued at the distributed ACD sites and thereby has the capability to track order of service and how close it is to FIFO service. Maintaining this virtual FIFO queue requires two simple measures which almost all ACDs can provide: the number of calls in queue and the oldest call waiting time.

Among the advantages of this new method are the following: 1) within the limitation of the information delay and the update interval, the CRP has knowledge of every call in queue, its current waiting time, and the site where it is located, 2) the procedure is self-correcting and errors cannot accumulate, 3) it provides ways of measuring the deviation from strict FIFO order of service, 4) it enables the estimation of the call removal rate from each queue 5) it allows for the use of the delay predictor/corrector method and 6) it allows the capability to requeue or transfer a call based on deviation from true FIFO order of service. This virtual FIFO method, also referred to herein as the simulated FIFO method, predicts the call removal rate (also referred to as the queue advance rate) at each distributed site, the differences in actual and estimated delays and other useful information.

Two simple data elements are required which are provided by the different ACDs manufactured by the various manufacturers mentioned above: the number of calls in queue (also used in the agent-pool-based approach) and the oldest call waiting time (how long has the oldest call in the queue been waiting). The simulated FIFO approach of the present invention is easy to establish in an existing system and is more robust to changes in customer operations. Moreover, as will be further described herein, the method and apparatus of the present invention may consider real-time data.

Referring to Figure 1, the simulated FIFO algorithm is applied at a CRP processor 125. CRP processor 125 in one embodiment of the present invention comprises TOPMS 102. In this embodiment, central controller 120 comprises CRP processor 125 and TOPMS 102. The virtual queue may be maintained in the form of a linked list in memory

The availability or busy status of any ACD agent $x_1, y_1, z_1 \dots z_m$ may be reported via the TOPMS. The drawing is simplified to suggest that m agents are on duty at each ACD location, when in fact the number of agents at each site typically differs. Availability and busy status of home agents 110 is reported via the network to system 120. This data reporting is shown, for example, in Figure 1 by way of link 116 to one of the ACD locations, which typically would be the location nearest the home agents 110. On the other hand, non-local (1-800, 1-900 and 1-888) calls are routed to the home agents upon command of the CRP 125 via Network Control Point (NCP) 100 for controlling the toll network. The CRP 125 is a typical processor comprising application algorithms including the agent-pool based call routing algorithm and memory for storing call traffic data updates forwarded via TOPMS 102.

A major issue associated with agent-pool-based algorithms is that the number of ACD agents $x_1, y_1, z_1 \dots z_m$ may be inaccurate due to sharing of agents across queues of ACDs and/or applications. This can be seen from Figure 2. For example, an airline agent may take international reservations A, domestic reservations B, first class/frequent flyer reservations C and further arrange for air freight D. Let us assume that D is a higher priority than A, C has the same priority as A and B has a lower priority than A. Four groups of agents are shown G1 - G4 each having a different specialty. G1 handles international reservations (A) while G2, for example, handles international (A) and domestic (B). Calls for international reservations can be queued in any serving line G1-G4 and when handled by one server, the call disappears from all the queues. Sharing of servers across queues complicates the problem that has to be solved by the agent-pool-based algorithm. Thus, the major issue in predicting expected delay with the agent pool based algorithm is in estimating the total agent pool size when calls can be queued among multiple queues and specialty queues as shown in Figure 2, thus introducing ambiguities.

Consequently, there remains a need in the art for a more simplified method and apparatus for dynamically routing calls in a distributed ACD environment. The agent-pool-based method is subject to the above-described inaccuracies and ambiguities, relies on the collection of too many data elements and relies on the availability of such data from different vendor ACDs. Moreover, a solution should consider both real-time and

historical data if available or collectible from the ACDs or home agents.

SUMMARY OF THE INVENTION

The problems and related problems of prior art methods of dynamically routing calls in a distributed queuing environment are solved by the principles of the present invention, a method for simulating a central FIFO queue (a virtual FIFO queue), whereby the system maintains a central queue of all the calls physically queued at the distributed ACD sites and thereby has the capability to track order of service and how close it is to FIFO service. Maintaining this virtual FIFO queue requires two simple measures which almost all ACDs can provide: the number of calls in queue and the oldest call waiting time. Among the advantages of this new method are the following: 1) within the limitation of the information delay and the update interval, the CRP has knowledge of every call in queue, its current waiting time, and the site where it is located, 2) the procedure is self-correcting and errors cannot accumulate, 3) it provides ways of measuring the deviation from strict FIFO order of service, 4) it enables the estimation of the call removal rate from each queue 5) it allows for the use of the delay predictor/corrector method and 6) it allows the capability to requeue or transfer a call based on deviation from true FIFO order of service. This virtual FIFO method, also referred to herein as the simulated FIFO method, predicts the call removal rate (also referred to as the queue advance rate) at each distributed site, the differences in actual and estimated delays and other useful information. Two simple data elements are required which are provided by the different ACDs manufactured by the various manufacturers mentioned above: the number of calls in queue (also used in the agent-pool-based approach) and the oldest call waiting time (how long has the oldest call in the queue been waiting). The simulated FIFO approach of the present invention is easy to establish in an existing system and is more robust to changes in customer operations. Moreover, as will be further described herein, the method and apparatus of the present invention may consider real-time data.

Referring to Figure 1, the simulated FIFO algorithm is applied at a CRP processor 125. CRP processor 125 in one embodiment of the present invention comprises TOPMS 102. In this embodiment, central controller 120 comprises CRP processor 125 and TOPMS 102. The virtual queue may be maintained in the form of a linked list in memory

of CRP processor 125 and may be dependent on pre-bundled update data from a TOPMS system 102 or upon real-time event-based data forwarded directly to the CRP 125 by the distributed ACDs 108₁, 108₂, . . . 108_n. The availability or unavailability or busy status of an agent x₁, y₁ . . . z_m may be reported directly to the TOPMS 102 or to the central controller 120 including TOPMS 102 via the ACDs. Home agent status may be reported via the network control point 100. Call removal rate, the rate at which calls are cleared from the queue, is an important component of the distributed site status predictions and several approaches for estimating call removal rate in a more efficient manner than prior art approaches will be discussed further herein. The three measures required for estimating call removal rate are number of calls in queue, waiting time of the oldest call in queue and number of available (idle) agents. Only the first two measures are required for maintaining the simulated FIFO queue of the present invention. The third measure is required for routing calls when agents are idle and for an efficient estimate of call removal rate.

These and other features of the present invention will become clear from reading the following detailed description of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 depicts, in simplified form, an exemplary telephone network comprising distributed ACDs 108 having agents x₁, y₁ . . . z_m of a distributed ACD system 115 including home agents 110 and embodying the principles of the present invention whereby a simulated FIFO approach to dynamic call processing according to the present invention is implemented in an algorithm applied at a processor and memory at a customer routing point (CRP) 125 from data collected via a system (TOPMS) 102 coupled to each of the distributed ACDs 108 or via event-based reporting directly by the ACDs to the CRP, where the home agent status is reported via NCP 100. While shown in the '744 patent as separate systems, the CRP 125 and TOPMS 102 may comprise central controller system 120 and be at the same location or distributed or one may share or be combined in functionality with the other.

Figure 2 exemplifies a problem of estimating agent pool size with prior art approaches to dynamic call routing and load balancing a distributed ACD system whereby

a call may be placed in multiple queues to be served by different agent groups or, in other words, servers are shared across queues of calls. This is commonly referred to as a multi-group queuing feature of ACDs.

Figures 3-6 describe an example of applying the simulated FIFO approach to a simple system where there are two distributed ACD sites, wherein a first procedure describes what happens when a new call arrives and a second procedure describes when a site update arrives; Figure 3 describes a virtual queue at time $T = 60$ in said first call arrival procedure; Figure 4 shows the same virtual FIFO queue after a new call arrives at $T = 61$; Figure 5 shows a virtual FIFO upon arrival of a site update according to said second site update procedure; and Figure 6 shows a call arrival after a site update.

Figures 7 and 8 relate to a simulation study performed to confirm the applicability the methods described herein versus perfect data reporting. Figure 7 describes the parameters used for each of five ACDs and update interval. Figure 8A is a key figure for describing the graphical data of Figures 8B-8F. Figures 8B-8F show the modeling results for ACDs of varying sizes.

DETAILED DESCRIPTION

Referring to Figure 1, there is shown a typical distributed ACD environment in which calls placed by callers, CALLER 1 . . . N, are served at Automatic Call Distributor (ACD) switch locations $108_1, 108_2, \dots, 108_n$ that may be dispersed all across the country, or the world, probably, in several different time zones. Serving agents $x_1 \dots x_m$ are located at a first automatic call distributor 108_1 ; agents $y_1 \dots y_m$ at a second ACD 108_2 ; and agents $z_1 \dots z_m$ at a last ACD in the system 108_n . The use of the term m should not be construed to limit the application of the invention to a situation where each ACD has m agents when typically the number of agents is different at each ACD site. Also, there may exist home agents 110 operating out of their homes such as home agent h_1 . Link 116 is shown as a typical link whereby data reporting may occur for home agents via their nearest ACD site. Calls as they are received into the system are distributed to the optimum site as a function of site status prediction (queue size and delays based on queue removal rate) and customer preferences. Of course, if there are many servers and few calls, as soon as a call comes in, they may be immediately served. Problems arise when

the calls cannot be distributed evenly among agents at different ACDs. Some callers will have to wait longer than other callers at different ACD or home agent sites if one site receives more calls than another site. Callers where the queue is long will complain about long waiting times and that there are not enough agents to serve their needs. According to the present invention, a method of distributing calls to load balance the system 115 is applied at a processor at a customer routing point 125 from site update data collected via a system known as TOPMS 102 or from real-time, event-based data reported directly to the CRP 125 by the ACDs or the home agents, the latter via NCP 100. The approach to dynamic call routing of the present invention is referred to herein as a simulated FIFO approach whereby a virtual queue is established at the CRP 125 which simulates how calls are handled at the distributed sites $108_1, 108_2, \dots, 108_n$ and 110. It is anticipated that central ACD controller system 120 may collect and utilize real time event-based data in conjunction with historical data as will be further described herein.

A detailed discussion of call processing in the environment of Figure 1 is found in a discussion of Figure 2 of U.S. Patent No. 5,530,744. For example, 1-800, 1-888, and 1-900 calls received at a terminating office of a local exchange carrier 106 from, for example, CALLER 1 are processed by routing signaling information to a network control point 100 and then to a CRP 125 which decides, based on load balancing criteria, the location of a next available agent and returns routing information to route the call to the site $108_1, 108_2, \dots, 108_n$ or home agent 110, whichever the next available agent may be expected.

In theory, real FIFO (First In First Out) service may be thought to be the most efficient call routing method. It provides the lowest average speed of answer (also referred to as queuing delay, call waiting time, and such) and is believed to have the smallest tail of the waiting time distribution compared with any other routing method. Real FIFO has two very attractive features for those that operate and maintain distributed call distribution environments such as that depicted in Figure 1: fairness to callers and efficiency in operation. FIFO is fair because calls are answered in FIFO order and each caller in the queue is served in turn. FIFO is efficient since agents cannot be idle while calls are waiting to be answered.

The performance of a system of distributed queues with a central virtual queue as discussed herein can be compared with a real FIFO queue where all agents of system 115 can serve the calls out of one queue. In practice, this arrangement is not used since agents would have to wait for calls to be sent out of the central queue. However, in theory, it serves as a bound on the best possible performance. Also, real FIFO service provides the lowest variance for the delay in queue. It also has the appeal of fairness in serving calls in order of arrival. While the simulated FIFO of the present invention cannot guarantee FIFO order of service, it provides a method of measuring and tracking deviation from FIFO order of service. This is advantageous since it enables a control that moves a call from a queue at one site to an idle server at another site if the deviation from FIFO becomes too large.

The simulated or virtual FIFO approach to call routing in a distributed ACD environment will now be discussed by way of example with reference to Figures 3-6.

The Simulated or Virtual FIFO Queue Approach to Call Routing

The simulated or virtual FIFO queue is maintained in memory at the processor of customer routing point CRP 125 as a linked list. As described above the processor may be a system processor 120 or distributed at CRP 125 and/or TOPMS 102. An entry for each new call routed by the CRP 125 is added to the tail of this virtual queue. Figure 3 shows an exemplary virtual queue comprising eight entries in a linked list. Each entry consists of a tabular listing of elements in memory of a processor at system 120. Each entry (for example, entries 1-8 of FIG. 3) in the table may consist of the following items, the last two of which are optional and not shown in FIG. 3:

<i>call_id</i> :	a call identifier, typically a sequential number identifying a call,
<i>time_arr</i> :	the call arrival time of the call identified by <i>call_id</i> ,
<i>site</i> :	the site to which the call is routed,
<i>ans_bit</i> :	answer bit representing whether the call has been answered or not, initially set to 0, and
<i>req_bit</i> :	requeue bit representing whether the call is requeued at another site, initially set to 0,
<i>delay</i> :	expected delay as calculated by the method described herein, in the

'744 patent or other method known in the art.

The last two variables are not essential for maintaining a virtual FIFO queue according to the present invention. On the other hand, expected delay may be useful for measuring the difference between the actual and estimated delays and using this as a correction factor as will be explained further herein.

The answer bit is changed to 1 when a call is no longer in queue at the serving sites. A call is removed from the queue either because it is answered or because the caller abandoned the call before the call is answered. The requeue bit is set to 1 if the call is requeued from one site to another. An entry is removed from the head of the queue if the answer bit is 1. (In Figures 3-6, entry 2 has an *ans_bit* set to 1 and is eventually deleted from the exemplary linked list). The next entry is also removed from the queue if its answer bit is 1 and the process continues until the head of the virtual FIFO queue is a call with the answer bit 0 or the virtual FIFO is empty (all calls have been served, and there are no calls waiting). This may be seen in Figures 5 or 6.

Two update procedures are performed to maintain the virtual queue:

Procedure 1: Each time a call is routed by CRP 125, create an appropriate entry with all the items identified above and add it to the tail of the virtual FIFO queue.

Procedure 2: Each time TOPMS 102 provides a site update to CRP 125, determine which calls have been answered or abandoned, change the answer bits for these calls to 1 and remove all calls with answer bits of 1 from the head of the virtual FIFO queue. The same procedure would follow without TOPMS except more frequently if event-based reporting is permitted directly by ACDs 108 and home agents 310 to CRP 125. See Figures 3-4 for an example of Procedure 1.

There are two ways to accomplish Procedure 2. One is by obtaining complete details of the call, for example, via real-time event based data reporting by ACDs. The other way is by obtaining oldest call waiting time. One can use the oldest call waiting time to infer which calls have been answered at a site as follows. See Figures 5 and 6 for an example of Procedure 2, using oldest call waiting time.

Suppose that an update from site *j* arrives at CRP 125 at time *T* and contains the following information:

NQ_j : represent the number of calls in the virtual queue with site identifier j ;
 ocw_j : be the waiting time of the oldest call in queue at site j ; and
 AAV_j : be the number of available (idle) agents at site j .

Only the first two variables are needed for maintaining the simulated FIFO queue.
 If no calls are in queue, then both of them have the value 0. The oldest call waiting at the
 site queue arrived at time $T - ocw_j$. Hence, all calls, if any, in the virtual FIFO with site
 $= j$ and with $time_arr$ less than $T - ocw_j$ are no longer in queue. The third variable, AAV_j ,
 is required for routing calls when agents are idle.

Referring to Figure 3, the entries of the linked list table of processor memory of
 CRP 125 comprise the parameters defined above and the present tail of the linked list is
 entry 8 at time $T = 56$ seconds. The depicted example represents data for just two ACD
 sites, *site 1* and *site 2*. Entries 2, 5 and 8 are for calls sent to site 2 and the remaining
 entries are for calls sent to site 1 by CRP 125. Note that the answer bit for entry 2 at site
 2 has been set to 1. This call has been answered. However, this entry has not been
 deleted from the virtual FIFO as it is not at the head of the virtual FIFO linked list.

Figure 4 shows the addition of a new tail where a new call arrives at $T = 61$
 seconds as entry 9. Entry 2 is still not the head of the linked list and so has not been
 deleted. Figures 5-6 will be discussed subsequently herein.

Not all calls must be routed via the CRP 125. Typically, referring to Figure 1,
 callers may connect to their local serving ACD 108 by dialing a number through their local
 exchange carrier 106. None of these calls are routed through the CRP 125 (which
 typically handles toll-free calls from more distant locations). A fixed fraction of calls may
 be routed directly to the primary site via NCP 100 without going through the CRP 125.
 All non-CRP calls are routed without the knowledge of the CRP 125. The CRP 125 has
 no knowledge of the number and arrival times of non-CRP calls.

The site updates include the total number of calls queued at the site, and the CRP
 125 has no way of knowing how many of the total number of calls are non-CRP calls and
 how many were routed by the CRP. Non-CRP calls can be accounted for in the virtual
 FIFO queue procedure in the following way:

After a site update has been processed as described above, let

nq_j : number of calls in the virtual FIFO with $site=j$ and $ans_bit=0$.

Then, if $nq_j < Nq_p$, that is, there are fewer calls in the virtual queue than at the site, then the difference is assumed to be due to non-CRP call arrivals at the site. Hence, $Nq_j - nq_j$ non-CRP calls must have arrived at the site j in the last update interval. So these calls can be inserted into the virtual FIFO queue with an expected arrival time $T - .5 \gamma$ if the previous site update was at time $T - \gamma$. If there is more than one non-CRP call, their arrival times may be equally spaced between $T - \gamma$ and T . The actual arrival time need not coincide with the expected arrival time and so the arrival time can be in error. The size of the error is bounded by the update interval. With frequent updates, for example, every half to every five seconds directly from the ACD locations, the error is negligible.

Calls also can abandon (the caller gives up and hangs up) while in queue at site. This can further complicate the picture. The condition where the number of calls in the virtual queue exceeds the number of actual calls in queue at the sites indicates that calls have become abandoned. Since it is not known at the CRP 125 which calls have become abandoned, nothing can be done immediately at a CRP 125. However, calls which abandon will ultimately show up as answered and be removed from the virtual FIFO. Thus, abandoned calls do not create a problem by themselves but can effect the count of non-CRP calls. For example, a call can abandon while another non-CRP call may arrive. The count of calls in queue will not change at the CRP 125 and the virtual queue procedure will not detect that this has happened. However, the procedure maintains the correct number of calls in queue with a non-CRP arrival maintained as a call that did not abandon. The net effect is a different arrival time *time_arr* for this call. One way of minimizing this error is to access information on abandoned calls from the ACD 108 which is available from most if not all ACDs.

The discussion of the FIFO process will be continued with reference to Figure 5. At Time $T = 63$, a data update arrives to say that the number of calls in queue at site 1 is 4; also, the oldest call waiting has been waiting 17 seconds. The oldest call waiting could not have arrived before the earliest arrival time and so that the earliest arrival time = $63 - 17$ seconds or 46 seconds. So 46 seconds is the presumed latest *time_arr*. Calls with an

earlier *time_arr* have their *ans_bit* changed to one, if not already one, and deleted from the list if they are at the head of the virtual FIFO queue. Call 10 and call 15 must have been answered. Since entry 2 becomes the head of the linked list after entry 1 is deleted, it may now be deleted.

5 Now the number of entries with answer bit 0 and site identifier 1 equals 3 and since the number of calls in the virtual queue minus the calls in queue at site 1 is 1 ($4 - 3$), a non-CRP call must have arrived during the last update interval from site 1 and an entry must be created in the virtual queue. Assuming the length of the update interval, $L = 2$ seconds, the new entry's arrival time is set to $63 - 1 = 62$ seconds. Thus, the head of the
10 linked list in the virtual queue becomes *call_id* 17 and the tail was *call_id* 26 until the non-CRP call is added as entry 7. This result is shown in Figure 6.

 In Figure 6, if a new call 28 arrives at time $T = 63$ for, for example, distribution to site 2, it would be added to the tail of the linked list as a new entry 8 (not shown) and so the process continues. From this example, the reader may come to an appreciation of
15 how the simulated FIFO approach may be applied to distributing and accounting for calls at a CRP 125.

 It takes some time for calls to reach the sites after the CRP 125 commands that they be routed to the site, for example, signaling and switching delays in addition to call waiting in queue delays. It also takes some time to collect information via TOPMS and
20 to send updates from the sites to the CRP either with or without the buffering provided by TOPMS. The sum of these two delays may be quite small and be bounded between minimum d_{min} and maximum d_{max} values in seconds. Now, the update which arrives at the CRP 125 at time T contains site information collected no earlier than time $T - d_{max}$ for T in the previously described procedures. The only exception is the adding of non-
25 CRP calls described above. In calculating the earliest arrival time, we may substitute $T - d_{min}$ for T as we want to be sure that a non-CRP call did arrive.

 Also, the virtual FIFO procedure is a self-correcting procedure in that both non-CRP calls which may have been added in error and abandoned calls will be removed from the virtual FIFO queue as soon as the earliest arrival time becomes larger than their arrival
30 time. Thus, errors are transient and do not accumulate.

If there are calls in the middle of the virtual FIFO with an answer bit value of one, then the FIFO principle is being violated (that is, if an unanswered call remains in a linked list ahead of an answered call). For example, call number 14 (entry 2) in Figure 3 has been answered before call 10 (entry 1) in violation of the FIFO principle. Call 14 arrived 13 seconds later than call 10.

The differences in arrival times or relative queue positions can both be used as measures of the amount of deviation from true FIFO service. A large difference may signal the need for corrective action such as requeuing of calls at another site.

Using a formula for estimating expected delay or other methods known in the art, the expected delay on arrival for each call can be stored in the virtual FIFO. The current expected delay equals the time already spent in queue ($= T - time_arr$) plus the remaining expected delay. The latter can be obtained from the current position in the virtual FIFO and the expected delay formula. A current expected delay significantly larger than the expected delay upon arrival can also signal the need for possible corrective action.

One can also estimate the call removal rate from the virtual FIFO queue. The call removal rate is the rate at which calls are being cleared from the queue, that is, the calls that are answered at the sites or abandoned. The number of calls answered in each update interval is directly available from the virtual FIFO queue. Divided by the duration of the last update interval (L), it gives the estimated call removal rate over that last update interval. Estimating call removal rates over short time intervals can be error prone. If the number of calls answered is small, then the standard deviation of the estimated rate can be quite high. Thus, the call removal rate should be taken over several update intervals or smoothed over time.

Expert agents are agents that have expertise that are not necessarily shared by other agents. For example, from Figure 2, there are agents that may have expertise in air cargo while all agents can handle domestic reservations. Agents with multiple skills or having sites with multi-group queuing per Figure 2 is a specialized issue in call routing algorithms. Such features have the potential of creating situations where the actual delay is consistently higher or lower than the delay estimated by a delay estimation procedure known in the art. An example of such a situation is when agents from another application

back up the application under a next available agent control which may lead to an under count or over count of the agents staffing the application. With multi-group queuing features, it is difficult to make the agent count more accurate. On the other hand, if the difference between actual and estimated delays is measured and used as a correction factor, then, agent count can be made more accurate. This option may be referred to as the delay predictor/corrector method. The virtual FIFO provides both the estimated and actual delays for each call routed by the CRP 125. Hence, the virtual FIFO provides the information necessary to calculate the correction factor and change the delay estimate. Routing logic and call removal rate estimation will now be discussed in some detail.

Routing Logic and Call Removal Rate (CRR) Estimation

At each call arrival to the CRP 125, the routing decision is made based on site status evaluation (that is, the queue length and delays) of all sites and customer specified routing preferences. The site status is predicted based on instantaneous information of the number of calls in queue or total number of agents available (idle), as follows:

$$EWT_j = \frac{(NQ_j - AAV_j + 1)}{CRR_j} \quad (1)$$

Here, CRR_j denotes the call removal rate for site j . When all agents are busy at a site, the expression (1) represents the expected wait time of a new call, i.e., $EWT_j = E[\text{Wait_time} \mid \text{Queue_size} = NQ_j]$. For a site with at least one idle agent, the EWT expression (1) would be negative and would indirectly balance the site occupancies.

Once a call is routed to a site k , the site's predicted number of calls in queue, NQ_k , is incremented by 1 and stored with an appropriate time stamp.

The '744 Patent Method For Estimating Call Removal Rate

The '744 patent method assumes exponential service times and estimates CRR_j as a ratio of the number of agents ($\#Agents_j$) and average call handling time (AHT_j). Numerous simulation studies based on customer data have evaluated the performance of this method. The results, as well as real experience with big customers, demonstrate that the routing algorithm can achieve performance benefits equal or close to a single pool of agents, which is used as a benchmark for the best possible outcome. This method is based

on at least seven site status inputs which are periodically preprocessed into required input parameters for the ARS routing logic.

Another Approach to Estimation Of Call Removal Rate

Now, we explore the approach for estimating call removal rate of the present invention by using the information provided by a virtual FIFO queue on the number of calls answered during the update interval. This novel approach assumes periodic updates of only three site specific measures: number of calls in queue, oldest call waiting time and total number of available agents. We need to estimate the maximum call answer rate, i.e., the answer rate when all agents are busy. However, a problem arises with periodic site status updates for those intervals when not all agents are busy. We need to determine which updates are valid to use. For each "chosen" interval, we calculate the answer rate and use exponential smoothing to estimate call removal rate.

We explore alternate approaches for estimating call removal rate CRR based on three Queue Advance Rate methods and one Call Completion Rate method.

Queue Advance Rate Methods, which are employed by some ACDs, calculate a new CRR as a rate at which queue advances when all agents are busy. These methods differ in their approach to inferring which intervals satisfy the "busy agents" condition:

Queue Advance Rate 1	No idle agents in current and previous site status update;
Queue Advance Rate 2	No idle agents in previous site status update only;
Queue Advance Rate 3	Positive queue size in previous site status update & queue size that exceeds the number of calls answered.

For each method, whenever an update interval satisfies the method's condition, we calculate the answer rate for this interval and use exponential smoothing to estimate call removal rate. All other update intervals are ignored.

Call Completion Rate Method uses all update intervals. It calculates first the call completion rate, ccr_j , based on calls completed (exponential smoothing is used). CRR_j estimate is then obtained by inflating the ccr_j with a correction factor based on average

number of idle agents.

The Average Number Of Agents Available, AAV_j^{avg} , is estimated at each site status update for site j . Simple exponential smoothing with smoothing factor SF1 is used to obtain the new estimate:

$$AAV_j^{avg} = SF1 \cdot AAV_j + (1.0 - SF1) AAV_j^{avg}, 0.0 < SF1 \leq 1.0. \quad (2)$$

The next two steps are performed only if the time since the last calculation of the call removal rate exceeds prespecified time interval of DTIME seconds.

For a site j , we define

- 10 ANS_j cumulative counter of calls answered. This counter, initially set to zero, is reset after every calculation of call removal rate;
- Δt_j^{calc} time since the last calculation of the call removal rate;
- DTIME minimum time between calculations of call removal rate.

At each update for the site j , if $\Delta t_j^{calc} < DTIME$, the method ignores this new update (and its time stamp) and continued with the prediction from the last site status update. Otherwise, it predicts the new CRR_j in the following way.

1. It estimates the number of calls completed, NCC_j , based on number of calls answered and the current and last update of the number of agents available:

$$NCC_j = ANS_j - AAV_j^{last} + AAV_j^{curr}. \quad (3)$$

It calculates the call completion rate by expression (4) below:

$$ccr_j = \text{Max} \left\{ SF2 \frac{NCC_j}{\Delta t_j^{calc}} + (1 - SF2) ccr_j, 0.005 \right\}, 0.0 < SF2 \leq 1.0. \quad (4)$$

2. A new estimate for the Call Removal Rate, CRR_j , is obtained by $CRR_j = IF_j ccr_j$, where the inflating factor IF_j is given by

$$IF_j = \frac{ccr_j AHT_j + AAV_j^{avg}}{ccr_j AHT_j}. \quad (5)$$

In equation (5), AHT_j is a static value based on customer historical data.

This value, together with the effective number of agents, is also used to initialize the smoothing procedure given by equation (4).

Call removal rate obtained by this method represents an estimate for the mean maximum potential service rate, i.e., service rate when all agents are busy. It is approximated by the ratio of call answer rate ccr_j and site utilization ρ_j , which is the reciprocal value of IF_j . This is a method of estimating call removal rate by estimating agents busy.

One simplification of this method can be obtained by calculating call completion rate based on the number of calls answered. This approach modifies only the step 1. of the above procedure by substituting NCC_j in equation (4) with ANS_j . For the smoothing factors that optimize performance, we found out that there is hardly any difference between the call completion rate method and its simplification.

Performance Comparison

Figures 7 and 8 relate to a simulation study performed to confirm the applicability the methods described herein versus perfect data reporting. Figure 7 describes the parameters used for each of five ACDs and update interval. Figure 8A is a key figure for describing the graphical data of Figures 8B-8F. Figures 8B-8F show the modeling results for ACDs of varying sizes. Efficiency of these methods was evaluated by simulation study. We modeled five sites of various sizes (Figures 8B-8F) with exponential service times and Poisson arrivals to the CRP 125. Calls are routed based on perfect information of number of agents and average handling time. In parallel, call removal rate is calculated based on alternate methods and compared against call removal rate based on perfect information. We varied the site utilization, update frequency, time between smoothing calculations, smoothing factors values, initial values for call removal rates, etc.

Figures 8B-8F illustrate the CRR estimates based on the above methods against the actual rate based on perfect information on number of agents and average call handling time for sites of various sizes and site utilization of 95%.

The results show that the call completion rate method tracks "perfect" call removal rate much better than any of the queue advance rate methods. Similar results were

obtained for other scenarios and parameters. Dynamic call distribution based on this method should perform reasonably well, while at the same time, it should eliminate the need for complicated (and, yet, inaccurate) data collection.

5 Thus there has been described a simulated FIFO queue algorithm for determining load balancing in a distributed ACD environment. Other features and practices may become apparent to one of ordinary skill from the above discussion. Any United States patents referenced herein should be deemed to be incorporated by reference as to their entire contents for an appreciation of the present invention. The present invention should only be deemed to be limited in scope by the claims which follow.

WHAT WE CLAIM IS:

1 1. A method for load balancing calls in a distributed automatic call distributor system
2 comprising a plurality of automatic call distributors, the method comprising the steps of:
3 simulating a queue at a routing point, the queue representing the plurality of
4 queues at each of the distributed automatic call distributors; and
5 routing calls entering the simulated queue.

1 2. A method as recited in claim 1 comprising the preliminary step of:
2 receiving the number of calls in queue, the oldest call waiting time and the total
3 number of available agents from each automatic call distributor.

1 3. A method as recited in claim 2 wherein said receiving step comprises receiving
2 periodic data updates from each automatic call distributor.

1 4. A method as recited in claim 3 wherein said periodic data updates comprise data
2 representing number of calls in queue and oldest call waiting time.

1 5. A method as recited in claim 2 comprising the further step of:
2 estimating call removal rate at each automatic call distributor by calculating call
3 completion rate.

1 6. A method as recited in claim 5 wherein the step of calculating call completion rate
2 is responsive to calculating number of calls answered from said simulated queue.

1 7. A method as recited in claim 5 wherein said step of calculating call completion rate
2 comprises calculating the number of calls completed responsive to periodic data updates
3 on agents available and on a number of calls answered from said simulated queue.

1 8. A method as recited in claim 5 wherein said call removal rate estimation step
2 comprises the step of inflating call completion rate based on the average number of

3 available agents.

1 9. A method as recited in claim 8 wherein said inflating step is performed according
2 to an inflating factor, the inflating factor being based upon call completion rate, the
3 average number of agents available and average call handling time.

1 10. A method as recited in claim 1 wherein said step of simulating a queue comprises
2 the steps of:

3 creating an entry for the simulated queue for each new call that is routed; and
4 adding the entry to the tail of the simulated queue.

1 11. A method as recited in claim 2 wherein said receiving step comprises periodic
2 updating of the simulated queue for each automatic call distributor.

1 12. A method as recited in claim 11 wherein said calls are routed through a routing
2 point and comprising the further step of determining calls in said simulated queue not
3 routed through a routing point.

1 13. A method as recited in claim 1 wherein said routing step comprises routing calls
2 entering the simulated queue on the basis of expected wait time.

1 14. A method as recited in claim 1 wherein said calls are routed through a routing
2 point and said step of simulating a queue comprises the step of forming a list of entries for
3 each call routed at said routing point.

1 15. A method as recited in claim 13 comprising the further steps of measuring the
2 actual wait time via the simulated queue, calculating a difference between the measured
3 wait time and the estimated expected wait time and correcting the expected wait time
4 estimate accordingly.

1 16. A method as recited in claim 9 further comprising the step of determining expected
2 wait time based on number of calls in queue and call removal rate.

1 17. A method as recited in claim 9 wherein routing decisions when agents are available
2 are further based on the number of available agents and call removal rate.

1 18. A method as recited in claim 11 further comprising the steps of
 calculating a deviation from FIFO order of service.

1 19. A method as recited in claim 18 comprising the further step of requeuing a call at
2 another automatic call distributor.

1 20. A method as recited in claim 1 comprising the preliminary step of receiving event-
2 based data for calls in said system.

1 21. A method as recited in claim 20 wherein said receiving step comprises updating
2 of the simulated queue for each automatic call distributor.

1 22. A method as recited in claim 21 wherein said simulated queue is updated with data
2 for each call answered and each call not routed through a routing point.

1 23. A method as recited in claim 21 comprising the further step of:
2 estimating call removal rate at each automatic call distributor by calculating call
3 completion rate.

1 24. A method as recited in claim 23 wherein the step of calculating call completion
2 rate is responsive to calculating number of calls answered.

1 25. A method as recited in claim 23 wherein said event-based data comprises agents
2 available and said step of calculating call completion rate is performed periodically based

3 on said agents available and on said number of calls answered from said simulated queue.

1 26. A method as recited in claim 23 wherein said event-based data comprises agents
2 available and said call removal rate estimation step comprises the step of inflating call
3 completion rate based on the average number of agents available.

1 27. A method as recited in claim 26 wherein said inflating step is performed according
2 to an inflating factor, the inflating factor being based upon call completion rate, the
3 average number of agents available and average call handling time.

1 28. A method as recited in claim 21 wherein said calls are routed through a routing
2 point and comprising the further step of determining calls in said simulated queue not
3 routed through a routing point.

1 29. A routing point in a distributed automatic call distributor system comprising a
2 plurality of automatic call distributors, the routing point comprising:

3 a processor for routing calls according to a simulated FIFO queue and for
4 receiving the number of calls in queue, the oldest call waiting time and the total number
5 of available agents for each automatic call distributor; and

6 memory for storing a list of entries as said simulated FIFO queue for each call
7 routed by the routing point.

1 30. The routing point as recited in claim 29, said routing point receiving periodic
2 updates from a data collection system.

1 31. A routing point in a distributed automatic call distributor system comprising a
2 plurality of automatic call distributors, the routing point comprising:

3 a processor for routing calls according to a simulated FIFO queue wherein said
4 routing point receives data as event-based data for calls in said system; and

5 memory for storing a list of entries as said simulated FIFO queue for each call

6 routed by the routing point.

1 32. A method for simulating a queue at a routing point in a distributed automatic call
2 distributor system comprising a plurality of automatic call distributors comprising the
3 steps of:

4 receiving the number of calls in queue and the oldest call waiting time from each
5 automatic call distributor and

6 simulating a queue at a routing point, the queue representing the plurality of
7 queues at each of the distributed automatic call distributors.

1 33. A method as recited in claim 32 comprising the further steps of:

2 receiving the total number of available agents from each automatic call distributor
3 and

4 estimating call removal rate at each automatic call distributor from calculating call
5 completion rate.

1 34. A method as recited in claim 32 wherein said receiving step comprises receiving
2 periodic data updates from each automatic call distributor.

1 35. A method as recited in claim 33 wherein the step of calculating call completion
2 rate is responsive to calculating number of calls answered from said simulated queue.

1 36. A method as recited in claim 33 wherein said step of calculating call completion
2 rate comprises calculating the number of calls completed responsive to periodic data
3 updates on agents available and on number of calls answered from said simulated queue.

1 37. A method as recited in claim 33 wherein said call removal rate estimation step
2 comprises the step of inflating call completion rate based on the average number of agents
3 available.

1 38. A method as recited in claim 37 wherein said inflating step is performed according
2 to an inflating factor, the inflating factor being based upon call completion rate, the
3 average number of agents available and average call handling time.

1 39. A method as recited in claim 32 wherein said step of simulating a queue comprises
2 the steps of:

3 creating an entry for the simulated queue for each new call that is routed; and
4 adding the entry to the tail of the simulated queue.

1 40. A method as recited in claim 32 wherein said receiving step comprises periodic
2 updating of the simulated queue for each automatic call distributor.

1 41. A method as recited in claim 40 comprising the further step of
2 determining which calls have been answered from said simulated queue.

1 42. A method as recited in claim 40 comprising the further step of
2 determining calls not routed through a routing point.

1 43. A method as recited in claim 32 comprising the further steps of
2 calculating a deviation from FIFO order of service and
3 requeuing a call at another automatic call distributor.

1 44. A method as recited in claim 32 comprising the further step of measuring the
2 actual wait time via the simulated queue, calculating the difference between the measured
3 wait time and the estimated expected wait time and correcting the expected wait time
4 estimate accordingly.

1 45. A method as recited in claim 32 further comprising the step of forming a list of
2 entries for each call to be routed at a routing point.

1 46. A method as recited in claim 41 further comprising the steps of
2 calculating a deviation from first-in first out order of service.

1 47. A method for simulating a queue at a routing point in a distributed automatic call
2 distributor system comprising a plurality of automatic call distributors comprising the
3 steps of:

4 receiving event-based data for calls to said system and
5 simulating a queue at the routing point, the queue representing the plurality of
6 queues at each of the distributed automatic call distributors.

1 48. A method as recited in claim 47 wherein said event-based data receiving step
2 comprises updating of the simulated queue for each automatic call distributor.

1 49. A method as recited in claim 47 comprising the further step of determining calls
2 in said simulated queue not routed through a routing point.

1 50. A method as recited in claim 47 wherein said simulated queue is updated with data
2 for each call answered and each call not routed through a routing point.

1 51. A method as recited in claim 47 comprising the further steps of:
2 receiving event-based data for a count of the total number of available agents from
3 an automatic call distributor and
4 estimating call removal rate at the automatic call distributor from calculating call
5 completion rate.

1 52. A method as recited in claim 47 wherein said step of simulating a queue further
2 comprises the steps of

3 creating an entry for the simulated queue for each new call that is routed; and
4 adding the entry to the tail of the simulated queue said receiving step comprises
5 receiving periodic data updates from each automatic call distributor.

1 53. A method as recited in claim 51 wherein the step of calculating call completion
2 rate is responsive to calculating number of calls answered from said simulated queue.

1 54. A method as recited in claim 51 wherein said step of calculating call completion
2 rate comprises calculating the number of calls completed responsive to agents available
3 and on number of calls answered from said simulated queue.

1 55. A method as recited in claim 51 wherein said call removal rate estimation step
2 comprises the step of inflating call completion rate based on the average number of agents
3 available.

1 56. A method as recited in claim 55 wherein said inflating step is performed according
2 to an inflating factor, the inflating factor being based upon call completion rate, the
3 average number of agents available and average call handling time.

1 57. A routing point in a distributed automatic call distributor system comprising a
2 plurality of automatic call distributors, the routing point comprising:
3 a processor for forming a simulated FIFO queue and for receiving the number of
4 calls in queue and the oldest call waiting time for each automatic call distributor; and
5 memory for storing a list of entries as said simulated FIFO queue.

1 58. The routing point as recited in claim 57, said routing point receiving periodic
2 updates from a data collection system.

1 59. A routing point in a distributed automatic call distributor system comprising a
2 plurality of automatic call distributors, the routing point comprising:
3 a processor for forming a simulated FIFO queue, said routing point receiving data
4 as event-based data for calls in said system; and
5 memory for storing a list of entries as said simulated FIFO queue.

AMENDED CLAIMS

[received by the International Bureau on 24 August 1998 (24.08.98);
original claims 46-59 replaced by amended claims 46-60 (3 pages)]

1 46. A method as recited in claim 32 further comprising the step of estimating call delay on call
2 arrival in the simulated queue.

1 47. A method as recited in claim 41 further comprising the steps of
2 calculating a deviation from first-in first out order of service.

1 48. A method for simulating a queue at a routing point in a distributed automatic call
2 distributor system comprising a plurality of automatic call distributors comprising the steps of:
3 receiving event-based data for calls to said system and
4 simulating a queue at the routing point, the queue representing the plurality of queues at
5 each of the distributed automatic call distributors.

1 49. A method as recited in claim 48 wherein said event-based data receiving step comprises
2 updating of the simulated queue for each automatic call distributor.

1 50. A method as recited in claim 48 comprising the further step of determining calls in said
2 simulated queue not routed through a routing point.

1 51. A method as recited in claim 48 wherein said simulated queue is updated with data for
2 each call answered and each call not routed through a routing point.

1 52. A method as recited in claim 48 comprising the further steps of:
2 receiving event-based data for a count of the total number of available agents from an
3 automatic call distributor and

4 estimating call removal rate at the automatic call distributor from calculating call
5 completion rate.

1 53. A method as recited in claim 52 wherein said step of simulating a queue further comprises
2 the steps of

3 creating an entry for the simulated queue for each new call that is routed; and

4 adding the entry to the tail of the simulated queue said receiving step comprises receiving
5 periodic data updates from each automatic call distributor.

1 54. A method as recited in claim 52 wherein the step of calculating call completion rate is
2 responsive to calculating number of calls answered from said simulated queue.

1 55. A method as recited in claim 52 wherein said step of calculating call completion rate
2 comprises calculating the number of calls completed responsive to agents available and on number
3 of calls answered from said simulated queue.

1 56. A method as recited in claim 52 wherein said call removal rate estimation step comprises
2 the step of inflating call completion rate based on the average number of agents available.

1 57. A method as recited in claim 55 wherein said inflating step is performed according to an
2 inflating factor, the inflating factor being based upon call completion rate, the average number of
3 agents available and average call handling time.

1 58. A routing point in a distributed automatic call distributor system comprising a plurality of
2 automatic call distributors, the routing point comprising:

3 a processor for forming a simulated FIFO queue and for receiving the number of calls in
4 queue and the oldest call waiting time for each automatic call distributor; and
5 memory for storing a list of entries as said simulated FIFO queue.

1 59. The routing point as recited in claim 58, said routing point receiving periodic updates from
2 a data collection system.

1 60. A routing point in a distributed automatic call distributor system comprising a plurality of
2 automatic call distributors, the routing point comprising:

3 a processor for forming a simulated FIFO queue, said routing point receiving data as
4 event-based data for calls in said system; and
5 memory for storing a list of entries as said simulated FIFO queue.

FIG. 1

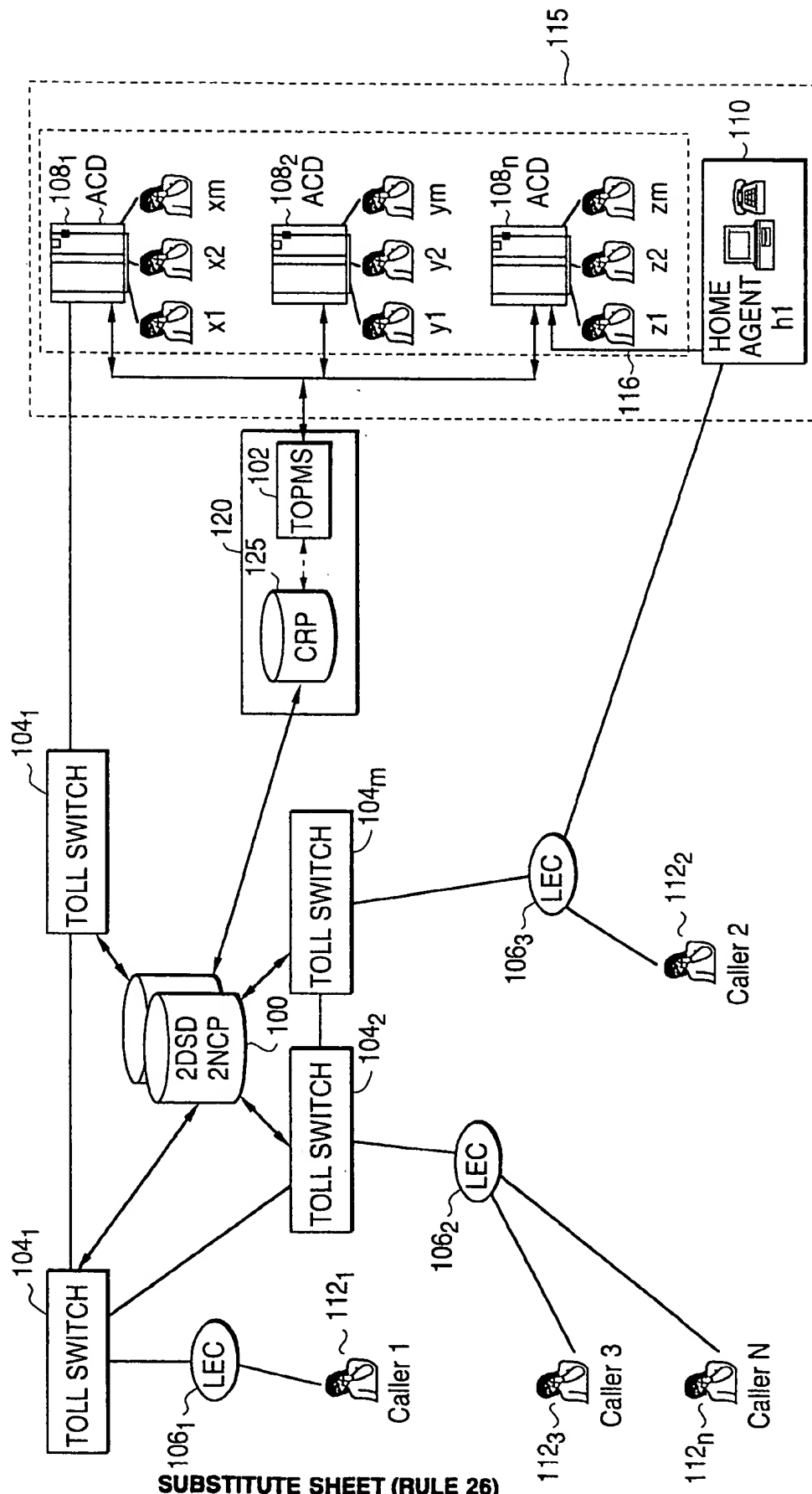


FIG. 2

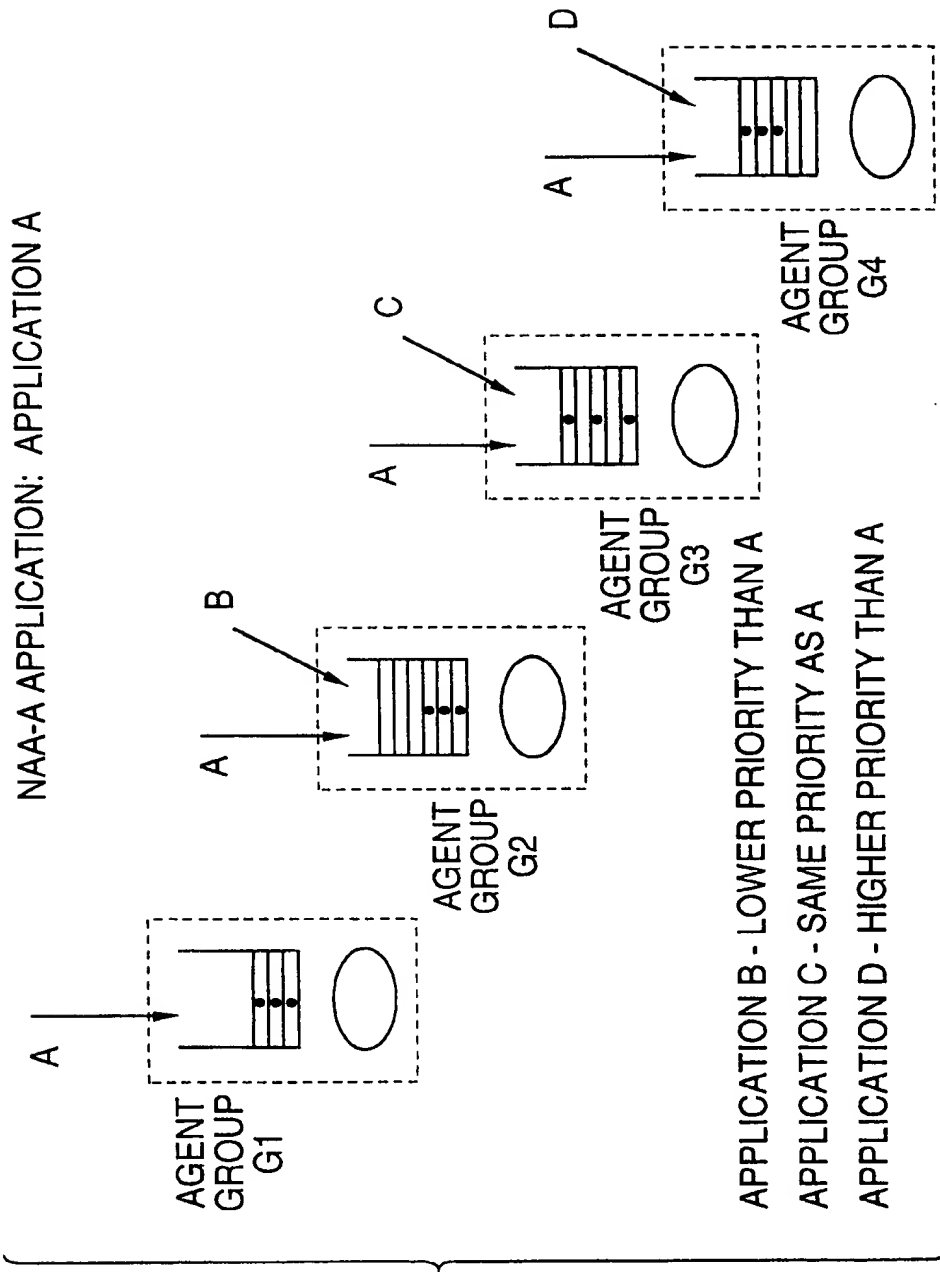


FIG. 3
MAINTAINED IN CUSTOMER ROUTING PROCESSOR AS A LINKED LIST...

LINKED LIST

position	entry	call_id	time_arr	site	ans_bit
head	1	10	30	1	0
	2	14	43	2	1
	3	15	44	1	0
	4	17	46	1	0
	5	18	50	2	0
	6	20	51	1	0
	7	22	55	1	0
	8	23	56	2	0
tail					

SIMULATED FIFO ENTRIES AT TIME 60.

FIG. 4
NEW CALL ARRIVES AT TIME 61 SECONDS AND IS ROUTED TO SITE 2.

LINKED LIST

position	entry	call_id	time_arr	site	ans_bit
head	1	10	30	1	0
	2	14	43	2	1
	3	15	44	1	0
	4	17	46	1	0
	5	18	50	2	0
	6	20	51	1	0
	7	22	55	1	0
	8	23	56	2	0
	9	26	61	2	0
tail					

SIMULATED FIFO ENTRIES AFTER CALL ARRIVAL.

5/8

FIG. 5

UPDATE FROM SITE 1: TIME = 63 WITH NQ = 4 AND ocw = 17 ->

EARLIEST ARRIVAL TIME = 63 - 17 = 46 SECONDS -->

position	entry	call_id	time_arr	site	ans_bit
deleted	-	10	30	1	1
deleted	-	14	43	2	1
deleted	-	15	44	1	1
head	1	17	46	1	0
	2	18	50	2	0
	3	20	51	1	0
	4	22	55	1	0
	5	23	56	2	0
	6	26	61	2	0
tail	7	27	62	1	0

SIMULATED FIFO ENTRIES AFTER CALL ARRIVAL.

6/8

FIG. 6

RESULTING SIMULATED FIFO LIST DISPLAYS 3 ENTRIES AT SITE 1:

TOPMS UPDATE SHOWS NQ = 4 -> 1 NON-CRP CALL ARRIVAL ->

NEW ENTRY 7

position	entry	call_id	time_arr	site	ans_bit
deleted	-	10	30	1	1
deleted	-	14	43	2	1
deleted	-	15	44	1	1
head	1	17	46	1	0
	2	18	50	2	0
	3	20	51	1	0
	4	22	55	1	0
	5	25	56	2	0
	6	26	61	2	0
tail	7	27	62	1	0

7/8

FIG. 7

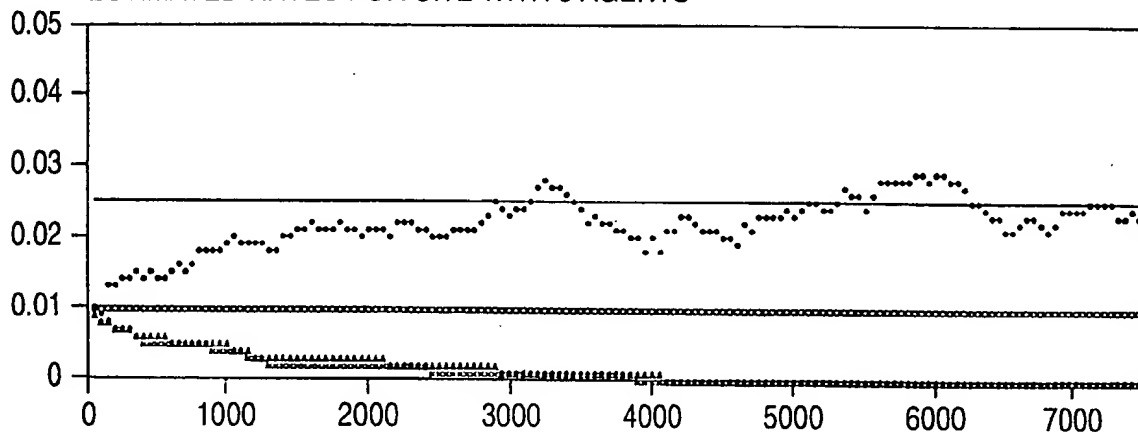
5.0 second updates					
Interval 1					
Agents	5	10	20	50	100
AHT	200	200	200	200	200
Arr. Int.	42.0	21.0	10.5	4.2	2.1
Util. %	95.2	95.2	95.2	95.2	95.2
AHT:	200.0				
DTime:	10.0				
Sm_Factors:	0.010	0.010	0.010		
Init rates:	0.010	0.030	0.060	0.100	0.300

FIG. 8A

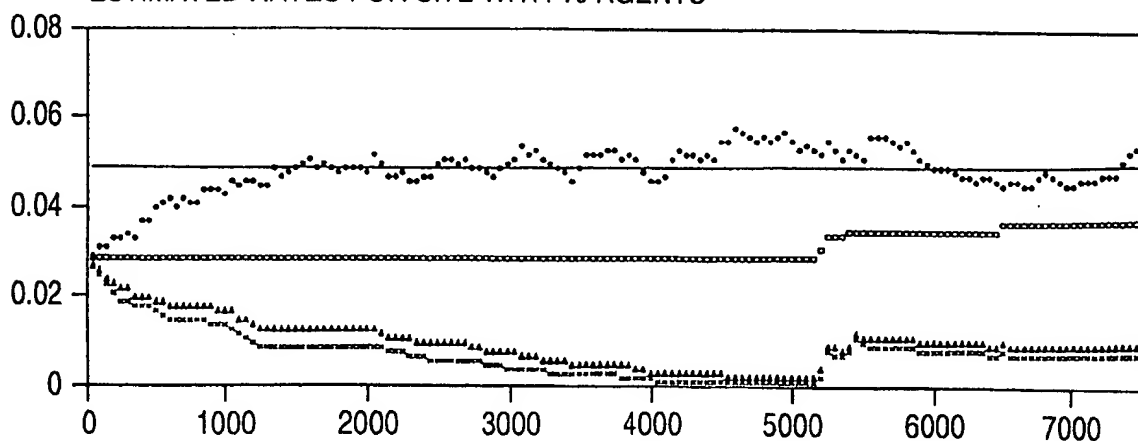
—	ACTUAL RATE			
•	CALL COMPLETION RATE		•	QUEUE ADVANCE RATE 1
•	QUEUE ADVANCE RATE 2		•	QUEUE ADVANCE RATE 3

FIG. 8B

ESTIMATED RATES FOR SITE WITH 5 AGENTS

**FIG. 8C**

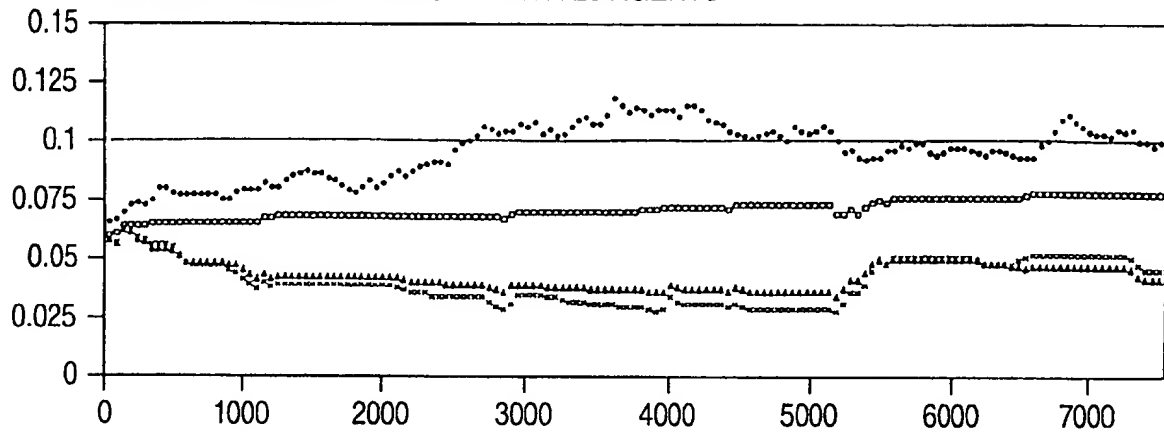
ESTIMATED RATES FOR SITE WITH 10 AGENTS



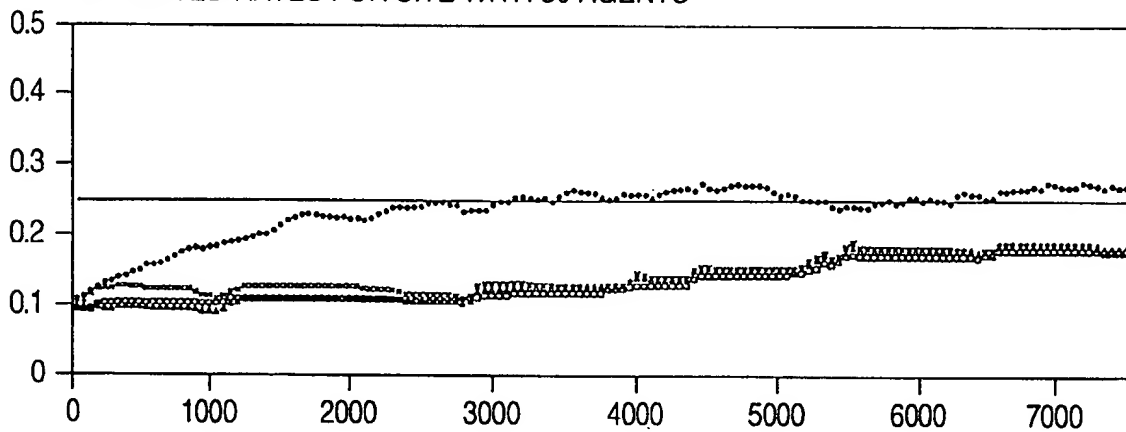
8/8

FIG. 8D

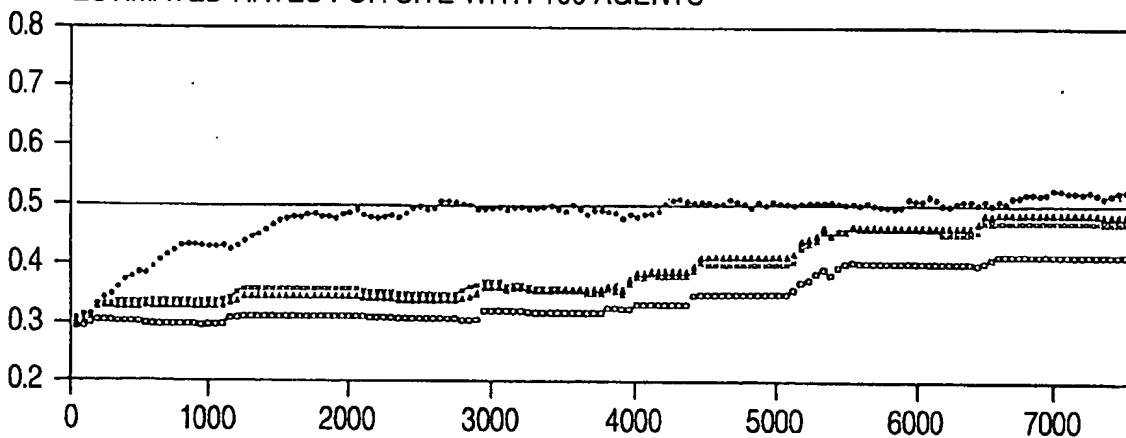
ESTIMATED RATES FOR SITE WITH 20 AGENTS

**FIG. 8E**

ESTIMATED RATES FOR SITE WITH 50 AGENTS

**FIG. 8F**

ESTIMATED RATES FOR SITE WITH 100 AGENTS



INTERNATIONAL SEARCH REPORT

national Application No
PCT/US 98/04872

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04M3/50

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>US 5 506 898 A (COSTANTINI RALPH J ET AL) 9 April 1996</p> <p>see figure 1 see column 2, line 2 - line 56 see column 3, line 2 - line 50</p> <p style="text-align: center;">--- -/--</p>	<p>1-7, 10, 11, 13, 15, 18, 29-32, 34, 39-41, 43, 45, 47, 50, 52, 57-59</p>

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

17 June 1998

Date of mailing of the international search report

23/06/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Weiss, P

INTERNATIONAL SEARCH REPORT

national Application No

PCT/US 98/04872

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 647 051 A (AT & T CORP) 5 April 1995 -----	1-4, 10, 11, 13, 14, 18-22, 28-32, 34, 39, 40, 42, 45, 47-52, 57-59

INTERNATIONAL SEARCH REPORT

Information on patent family members

national Application No

PCT/US 98/04872

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5506898 A	09-04-1996	NONE	
EP 0647051 A	05-04-1995	CA 2129942 A	31-03-1995
		JP 7170546 A	04-07-1995
		US 5633924 A	27-05-1997